|  |
| --- |
| [鍵入公司名稱] |
| Firebase Realtime Database |
| [鍵入文件副標題] |
|  |
| **Sonny.H.Shih** |
| **2017/7/20** |

|  |
| --- |
| [在此處鍵入文件摘要。摘要通常是文件內容的簡短摘要。在此處鍵入文件摘要。摘要通常是文件內容的簡短摘要。] |

目錄

[一、 簡介 2](#_Toc492628608)

[1. Firebase說明文件 2](#_Toc492628609)

[2. Firebase 內容 2](#_Toc492628610)

[3. Firebase Realtime Database 3](#_Toc492628611)

[3.1 說明 3](#_Toc492628612)

[二、 建立Realtime Database 5](#_Toc492628613)

[1. 在Firebase 上建立專案 5](#_Toc492628614)

[2. Android Studio 中的設定 8](#_Toc492628615)

[3. 設定Database 讀取權限 9](#_Toc492628616)

[3.1 Defalut設定 10](#_Toc492628617)

[3.2 Public設定 11](#_Toc492628618)

[3.3 User設定 12](#_Toc492628619)

[3.4 Private設定 13](#_Toc492628620)

# 簡介

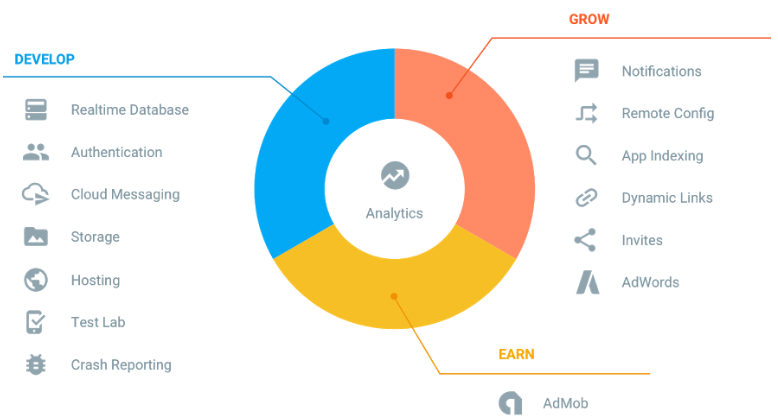
## Firebase說明文件

* URL: <https://firebase.google.com/>
* Firebase Console: <https://console.firebase.google.com/>

## Firebase 內容

Firebase提供了多種服務 (<https://firebase.google.com/features/>)

* Develop
* Grow
* Earn



Firebase主要核心是Firebase Analytics分析的功能。

## Firebase Realtime Database

參考: <https://firebase.google.com/docs/database/>

### 說明

* 使用 NoSQL Cloud database 做資料同步或是存儲資料
* 所有裝置可以realtime資料同步
* 裝置offline再online時，就可做資料同步
* 資料傳輸是以JSON的格式傳遞
* 可以在不同的平台使用。iOS, Android, Javascript。可同步更新資料。
* 可以結合Firebase Authentication 帳號的管理，開發者可以決定誰可以access什麼資料，如何access資料。
* Firebase Realtime Database 是NoSQL資料，它有不同於「關聯式資料庫」(Relational Database)有不同的優化與功能。
* 運作原理

透過帳號/密碼或是第三方的登入(Ex: FB, Twitter…)方式取得認証，然後把這認証傳給Firebase Authentication SDK，就會傳回後端，Firebase會檢驗這個認証回傳資料。驗証成功後，就可以取得/修改user的基本資料，且可以使用Firebase提供的服務與產品 (Ex: Realtime Database, Storage )。

# 建立Realtime Database

程式下載: <https://github.com/firebase/quickstart-android/tree/master/auth>

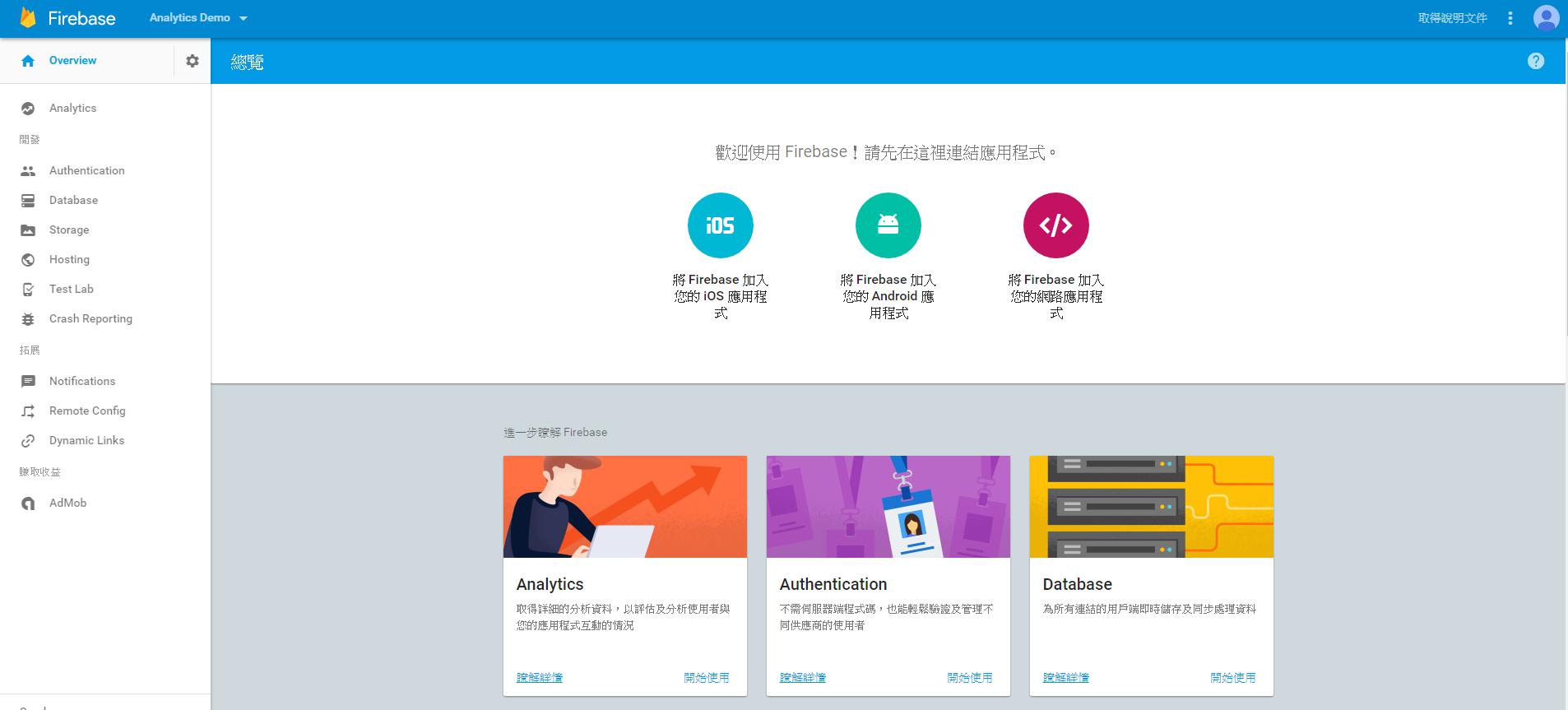
說明: <https://firebase.google.com/docs/auth/>

## 在Firebase 上建立專案

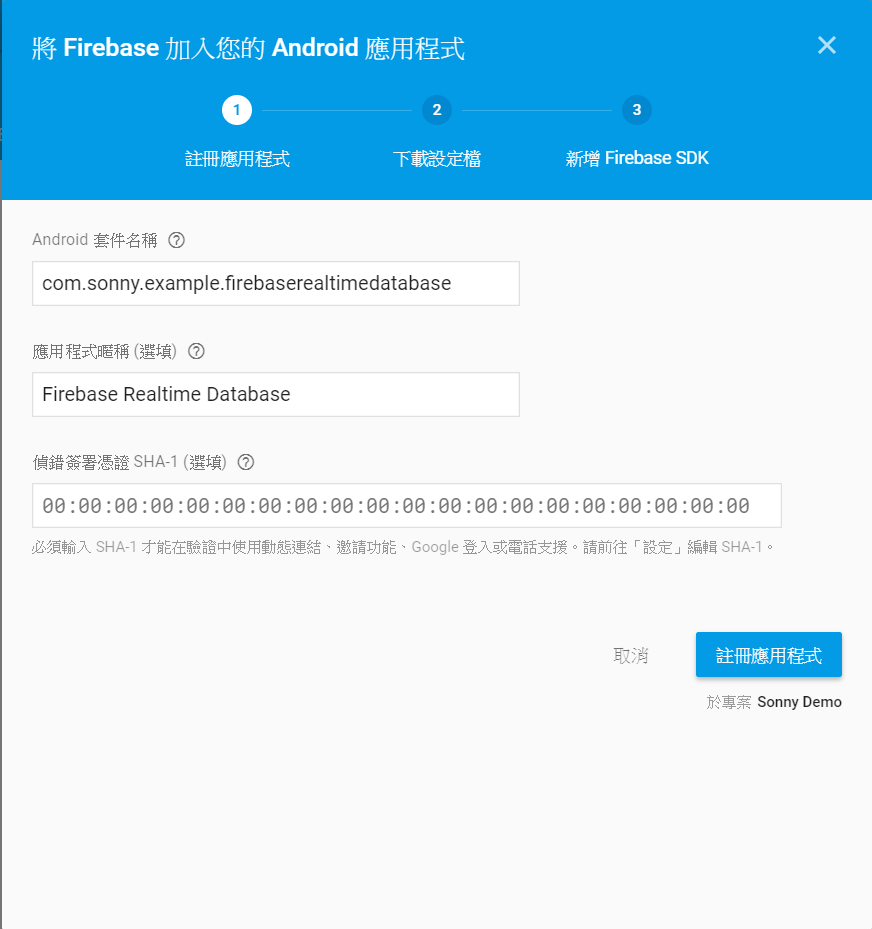
* 建立新專案

* 建立完成，選擇"將Firebase加入您的Android應用程式"

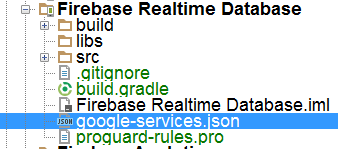


* 建立應用程式



* 下載google-services.json檔案，放到專案中





## Android Studio 中的設定

* 參考: <https://firebase.google.com/docs/auth/android/start/>
* 在 App 層級的build.gradle中加入

compile 'com.google.firebase:firebase-database:11.0.2'

如下圖



* 設定ProGuard 混淆器

在 proguard-rules.pro 中加入

# Add this global rule

-keepattributes Signature

# This rule will properly ProGuard all the model classes in

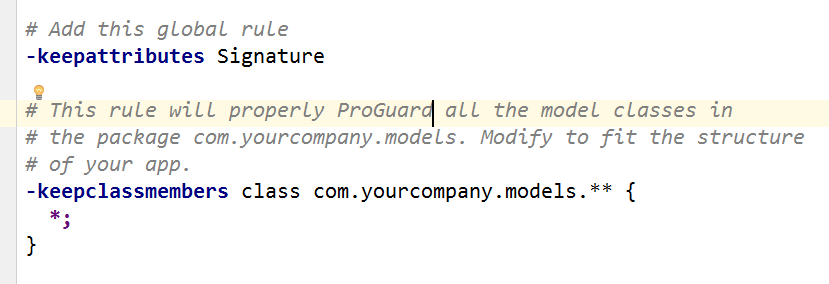
# the package com.yourcompany.models. Modify to fit the structure

# of your app.

-keepclassmembers class com.yourcompany.models.\*\* {

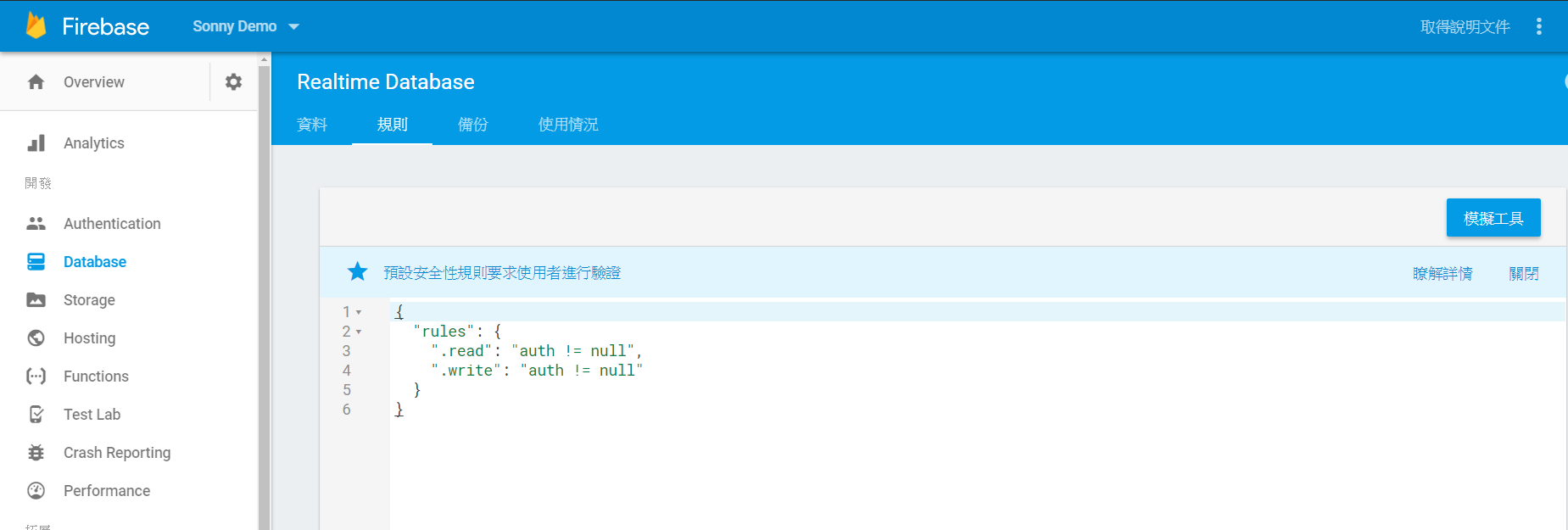
\*;

}



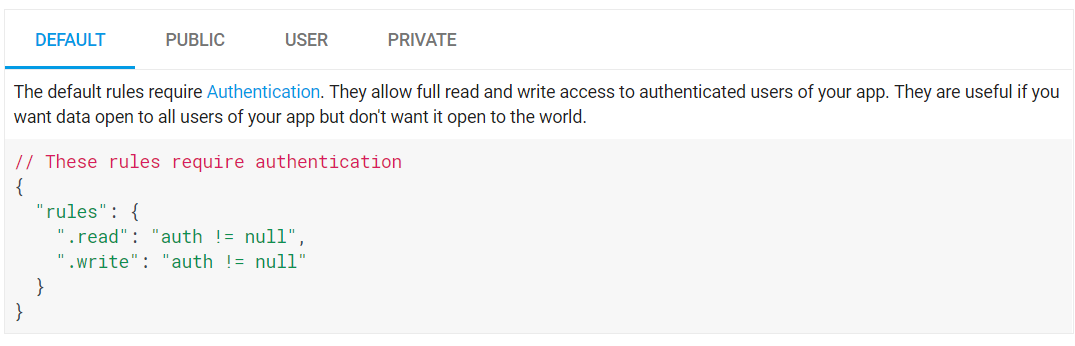
## 設定Database 讀取權限

* 參考: <https://firebase.google.com/docs/database/security/quickstart>



### Defalut設定

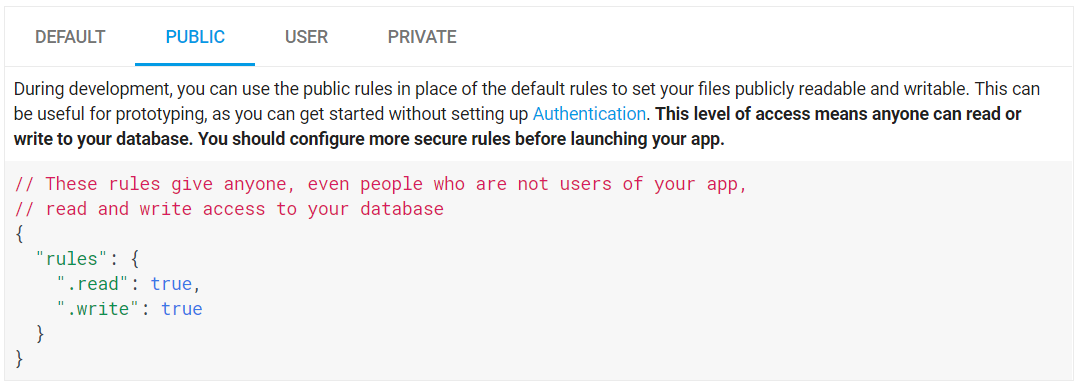
// These rules require authentication  
{  
  "rules": {  
    ".read": "auth != null",  
    ".write": "auth != null"  
  }  
}



**此設定需配合Firebase Authentication，使用者要先在Firebase Authentication註冊與認証，才有權限使用Realtime Database。**

### Public設定

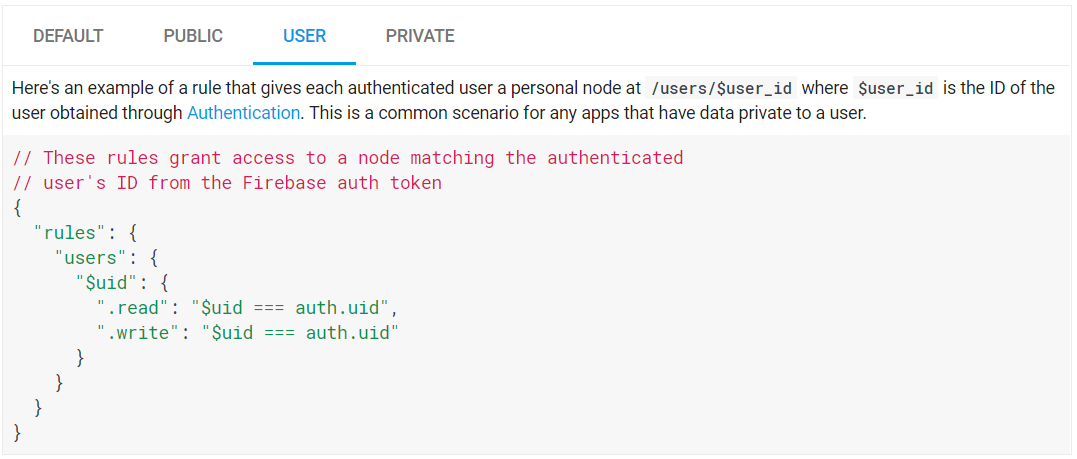
// These rules give anyone, even people who are not users of your app,  
// read and write access to your database  
{  
  "rules": {  
    ".read": true,  
    ".write": true  
  }  
}



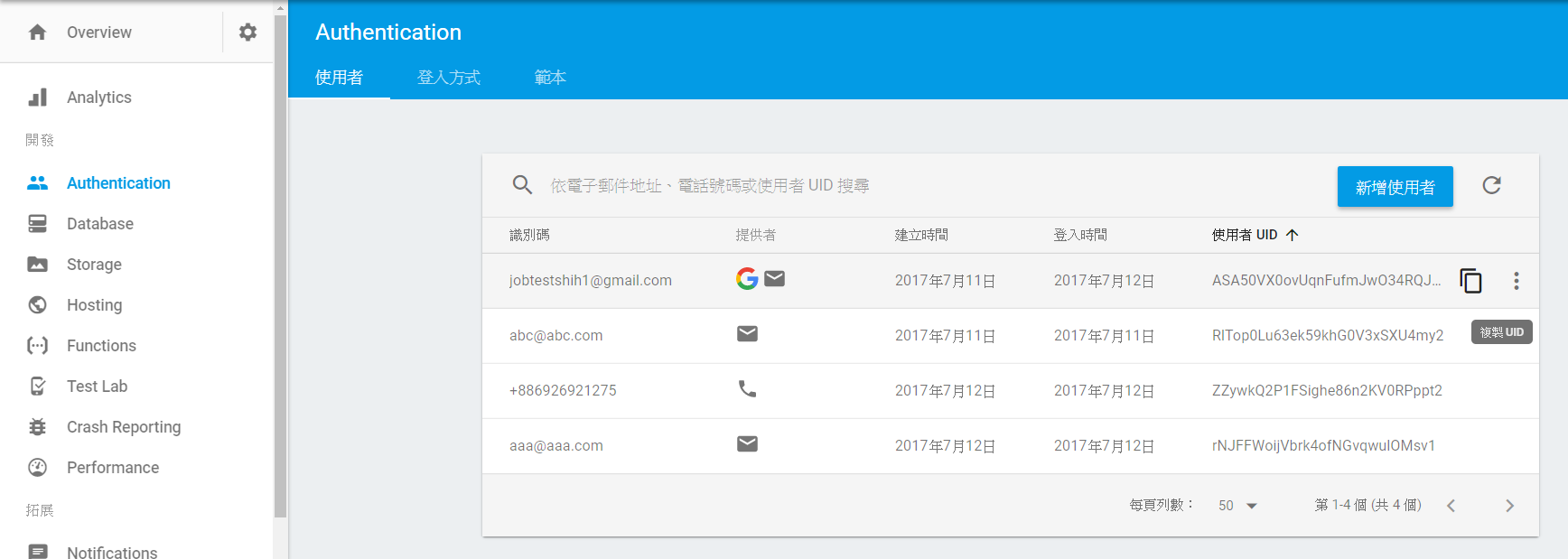
此設定不需要使用Firebase Authentication，任何人都可以使用Realtime Database。

### User設定

// These rules grant access to a node matching the authenticated  
// user's ID from the Firebase auth token  
{  
  "rules": {  
    "users": {  
      "$uid": {  
        ".read": "$uid === auth.uid",  
        ".write": "$uid === auth.uid"  
      }  
    }  
  }  
}

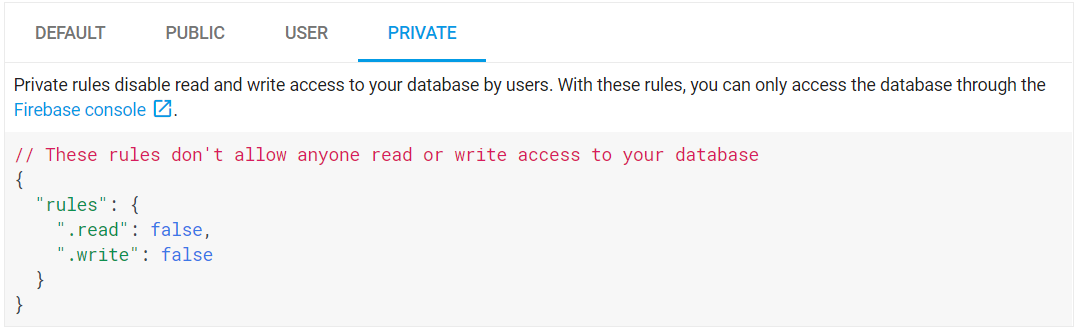


**此設定需配合Firebase Authentication，使用者要先在Firebase Authentication註冊與認証，且有被指定到的帳號才有權限使用Realtime Database。**



### Private設定

// These rules don't allow anyone read or write access to your database  
{  
  "rules": {  
    ".read": false,  
    ".write": false  
  }  
}



完全不給任何人讀寫Realtime Database。